Expert Systems with Applications 38 (2011) 14269–14275

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A two-stage framework for cross-domain sentiment classification

Qiong Wu, Songbo Tan *

Key Laboratory of Network, Institute of Computing Technology, Chinese Academy of Sciences, China

ARTICLE INFO

Keywords:
Sentiment analysis
Opinion mining
Information retrieval
Data mining

ABSTRACT

Supervised sentiment classification systems are typically domain-specific, and the performance decreases sharply when transferred from one domain to another domain. Building these systems involves annotating a large amount of data for every domain, which needs much human labor. So, a reasonable way is to utilize labeled data in one existed (or called source) domain for sentiment classification in target domain. To address this problem, we propose a two-stage framework for cross-domain sentiment classification. At the "building a bridge" stage, we build a bridge between the source domain and the target domain to get some most confidently labeled documents in the target domain; at the "following the structure" stage, we exploit the intrinsic structure, revealed by these most confidently labeled documents, to label the target-domain data. The experimental results indicate that the proposed approach could improve the performance of cross-domain sentiment classification dramatically.

1. Introduction

Different from traditional text classification (e.g. Tan, 2005, 2006; Tan, Cheng, Ghanem, Wang, & Xu, 2005), sentiment classification, whose goal is to determine the opinion (e.g., negative or positive) of a given document, has recently received a lot of attention in the Natural Language Processing (NLP) community.

In most cases, a variety of supervised classification methods can perform well in sentiment classification (Pang, Lee, & Vaithyanathan, 2002). But when training data and test data are from different domains, the supervised classification methods often cannot perform well. The reason is that training data do not have the same distribution with test data so test data could not share the information from training data. This is often found in the following two cases. One case is that there are different features in different domains. For instance, the word "warm-hearted" frequently appears in hotel reviews, but it hardly appears in electronics reviews. On the other hand, some features that have high correlations with certain class labels in the training domain do not have as high correlations with the same class labels any more in the new domain, and vise versa (Jiang & Zhai, 2007). For instance, the word "portable" may be positive in electronics reviews, but it means nothing in hotel reviews.

Therefore, the labeled data in the same domain with test data is considered as the most valuable resources for the sentiment classification. However, such resources in different domains are very imbalanced. In some traditional domains or domains of concern, many labeled sentiment data are freely available on the web, but in other domains, labeled sentiment data are scarce and it involves much human labor to manually label reliable sentiment data. So, the challenge is how to utilize labeled sentiment data in one domain (that is, source domain) for sentiment classification in another domain (that is, target domain). This raises an interesting task, cross-domain sentiment classification (or sentiment transfer). In this work, we focus on one typical kind of sentiment transfer problem, which utilizes only training data from source domain to improve sentiment classification performance for target domain, without any labeled data for the target domain.

Realizing the challenges posed by sentiment transfer, some researchers have explored a number of techniques to improve the performance of the sentiment transfer. However, the difficulties for sentiment transfer are as follows: the first one is that the distribution of the target domain is not same with that of the source domain, and hence we need to build a bridge to share the information got from the source domain; the second one is the intrinsic structure of the target domain is static, so we need to utilize the intrinsic structure collectively revealed by target domain. In brief, a good method for sentiment transfer is expected to utilize the information contained in the source domain as much as possible, and moreover, follow the intrinsic structure revealed by target domain as much as possible.

In light of the difficulties for sentiment transfer, we address the task of sentiment transfer via a "building a bridge, following the structure" two-stage framework. Specifically, at the "building a bridge" stage, we build a bridge between the source and the target domain by applying the *SentiRank* algorithm (which uses the accurate labels of source-domain documents as well as the "pseudo" labels of target-domain documents to label the target-domain

* Corresponding author. Address: Key Laboratory of Network, P.O. Box 2704, Beijing 100190, PR China. Tel.: +86 10 62600928; fax: +86 10 62600905.
E-mail address: tansongbo@software.ict.ac.cn (S. Tan).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.04.240

documents initially), and then choose some high-quality seeds from the target domain which are most confidently labeled. At the "following the structure" stage, we utilize the intrinsic structure (by employing the manifold-ranking process) to compute the sentiment score for every document that denotes the degree of sentiment orientation. So we can label the target-domain data based on these scores.

Our contribution is threefold. First, while existing sentiment-transfer approaches typically rely on a generative model, our approach build a bridge between two domains to get some high-quality seeds from the target domain, then follow the structure embodied by the seeds to improve the performance of sentiment transfer. Second, while existing manifold-ranking-based approaches typically start with manually labeled seeds, our approach relies only on seeds that are automatically extracted and labeled from the target domain. Third, we contribute by making use of the graph-ranking algorithm to get the opinions of the documents context-dependently.

The proposed approach is evaluated on three domain-specific sentiment data sets. The experiment results show that our approach can dramatically improve the accuracy when transferred to another target domain. And we also conduct extensive experiments to investigate the parameters sensitivity.

2. Related work

2.1. Sentiment classification

Sentiment classification could be grouped into word level, sentence level or document level. This paper focuses on document-level sentiment classification, and supervised classification methods are proved to be effective to address this problem.

Supervised classification methods (e.g. (Pang et al., 2002; McDonald, Hannan, Neylon, Wells, & and Reynar, 2007; Cui, Mittal, & Datar, 2006)) usually train a sentiment classifier on labeled data for sentiment classification task. It has drawn more and more attention because of its applications in many aspects. Pang et al. (2002) applied three traditional supervised classification methods to sentiment classification, and the experimental results showed that standard machine learning techniques definitively outperformed human-produced baselines. McDonald et al. (2007) investigated a structured model for jointly classification of reviews when varying the level of granularity. Cui et al. (2006) presented experimental results with different machine-learning algorithms over huge amount of online product reviews. The experimental results showed that a discriminating classifier combined with high order n-grams as features could achieve better performance.

However, supervised sentiment classification requires that labeled and unlabeled data should be under the same distribution, so that the classifier built by the labeled data could be well applied to the unlabeled data. But in sentiment transfer field, the labeled and unlabeled data are often from different domains, and often have different distributions. This is inconsistent with the basic requirements of supervised methods, thus this kind of effective methods cannot be directly used in cross-domain sentiment classification.

2.2. Semi-supervised learning

In practical text categorization, labeled documents are often very sparse while there are often abundant unlabeled documents. As a result, exploiting these unlabeled data has become an active research problem in text classification recently.

Nigam, McCallum, Thrun, and Mitchell (1998) introduced an EM-like approach that combines Expectation–Maximization (EM) algorithm with Naive Bayes classifier. In this algorithm, He first derived pseudo labels for unlabeled documents, and then incorporate these unlabeled data into supervised learning. This process will be iterated until convergence.

Following this direction, Lanquillon (2000) described a general framework for extending any text-learning algorithm to utilize unlabeled documents. In this framework, he also used an Expectation–Maximization-like scheme. In his work, he used three traditional methods, i.e., Naive Bayes classifier, Single Prototype (or Centroid) classifier, and SVM, as base classifier.

Blum and Mitchell (1998) proposed a Co-Training method that splits the original feature set into two conditional independent feature sets. The algorithm initially trains two classifiers separately based on labeled data, and then each algorithm's predictions on new unlabeled examples are used to enlarge the training set of the other. He also provided a PAC-style framework for the general problem of learning from both labeled and unlabeled data.

Joachims (1999) modified SVM to exploit the unlabeled data (often called TSVM). TSVM expects to find a low-density area of data and constructs a linear separator in this area so that the margin over both the labeled data and the unlabeled data can be maximized.

Chawla and Karakoulas (2005) presented an empirical study of various semi-supervised learning techniques on a variety of datasets. They also introduced two techniques from Econometrics, namely reweighting and bivariate probit, for semi-supervised learning. Meanwhile, they answered various questions that could be important to learn from labeled and unlabeled datasets.

Ando and Zhang (2005) presented a novel semi-supervised learning paradigm called structural learning. The method intended to find what good classifiers were like by learning from thousands of automatically generated auxiliary classification problems on unlabeled data. By doing so, the common predictive structure shared by the multiple classification problems could be discovered, and they could then be used to improve the performance of the target problem.

Ikeda, Takamura, and Okumura (2008) proposed a semi-supervised blog classification method. In this method, they assumed that entries from the same blog had the same characteristics. With this assumption, the proposed method captured the characteristics of each blog, such as writing style and topic, and used these characteristics to improve the classification accuracy.

2.3. Transfer learning

Transfer learning aims to utilize labeled data from other domains or time periods to help current learning task, and the underlying distributions are often different from each other.

In the past years, many researchers have been working on this field and have proposed many approaches, including classifier adaptation (Chelba & Acero, 2004; DaumeIII & Marcu, 2006), bridged refinement (Xing, Dai, Xue, & Yu, 2007), two-stage approach (Jiang & Zhai, 2007), consensus regularization framework (Luo, Zhuang, Xiong, Xiong, & He, 2008) and so on. Chelba and Acero (2004) presented a novel technique for maximum "a posteriori" (MAP) adaptation of maximum entropy and maximum entropy Markov models. DaumeIII and Marcu (2006) introduced a statistical formulation for domain adaptation in terms of a simple mixture model. They also presented an instantiation of this framework to maximum entropy classifiers and the linear chain counterparts. Xing et al. (2007) proposed a bridged refinement algorithm, which take the mixture distribution of the training and test data as a bridge to better transfer from the training data to the test data. Jiang and Zhai (2007) presented a two-stage approach to domain adaptation. At the first generalization stage, they looked for a set of features generalizable across domains, and at the second stage,

they picked up useful features specific to the target domain. Luo et al. (2008) proposed a consensus regularization framework where a local classifier was trained by considering both local data in a source domain and the prediction consensus with the classifiers from other source domains. Raina, Battle, Lee, Packer, and Ng (2007) presented a new transfer-learning framework called "self-taught learning". They used sparse coding to construct higher-level features over a large number of unlabeled data randomly downloaded from the Internet. These features formed a succinct input representation and significantly improved classification performance. Do and Ng (2005) proposed an algorithm for automatically learning the parameter function from related classification problems. The parameter function found by their algorithm then defined a new learning algorithm for text classification, which could be applied to novel classification tasks.

Recently, some researchers attempted to address sentiment transfer learning.

Some studies rely on only the labeled documents to improve the performance of sentiment transfer (e.g. (Aue & Gamon, 2005; Blitzer, Dredze, & Pereira, 2007; Tan, Wang, Wu, & Cheng, 2008, 2009; Dasgupta & Ng, 2009)). Aue and Gamon (2005) used four different approaches to customize a sentiment classification system to a new target domain using a small amount of labeled training data. Blitzer et al. (Blitzer et al., 2007) applied structural correspondence learning to automatically induce correspondences among features from different domains. Tan et al. (2008) used classifier trained in source domain to label some informative unlabeled examples in target domain, and trained the base classifier again. In another paper of Tan, Cheng, Wang, and Xu (2009), they proposed Frequently Co-occurring Entropy to pick out generalizable features that occurred similarly in both domains, and then proposed Adapted Naïve Bayes to train a classifier suitable for the target-domain data. Dasgupta and Ng (2009) proposed a semi-supervised approach to sentiment classification, where they first mined the unambiguous reviews using spectral techniques and then exploited them to classify the ambiguous reviews via a novel combination of active learning, transductive learning, and ensemble learning.

Moreover, some studies rely on only the sentiment words to improve the performance of sentiment transfer (e.g. (Gamon & Aue, 2005; Andreevskaia & Bergler, 2008)). Aue and Gamon (2005) proposed a method to automatically identify the sentiment vocabulary suitable for a given domain. Andreevskaia and Bergler (2008) presented a sentiment annotation system that integrated a corpus-based classifier trained on a small set of annotated in-domain data and a lexicon-based classifier trained on WordNet.

However, most of the existing studies rely on only utilizing the information from the source domain to address the task of sentiment transfer, while ignoring the intrinsic structure of the target domain.

In this paper, we design an algorithm for sentiment transfer by taking into account the relationship between source domain and target domain as well as the intrinsic structure of the target domain.

3. Our approach

3.1. Overview

As mentioned in the introduction, we employ a two-stage framework for sentiment transfer. The framework of the proposed approach is illustrated in Fig. 1.

The framework consists of two stages: a "building a bridge" stage and a "following the structure" stage. At the "building a bridge" stage, we (1) build a bridge between the source domain and the target domain with the help of a *SentiRank* algorithm to get the sentiment scores of the target-domain documents, (2) utilize the sentiment scores to identify a small number of most confidently labeled documents as high-quality seeds to embody the intrinsic structure of the target domain. At the "following the structure" stage, we (3) follow the structure of the target domain by applying a manifold-ranking algorithm; (4) use the manifold-ranking scores to label the target-domain data.

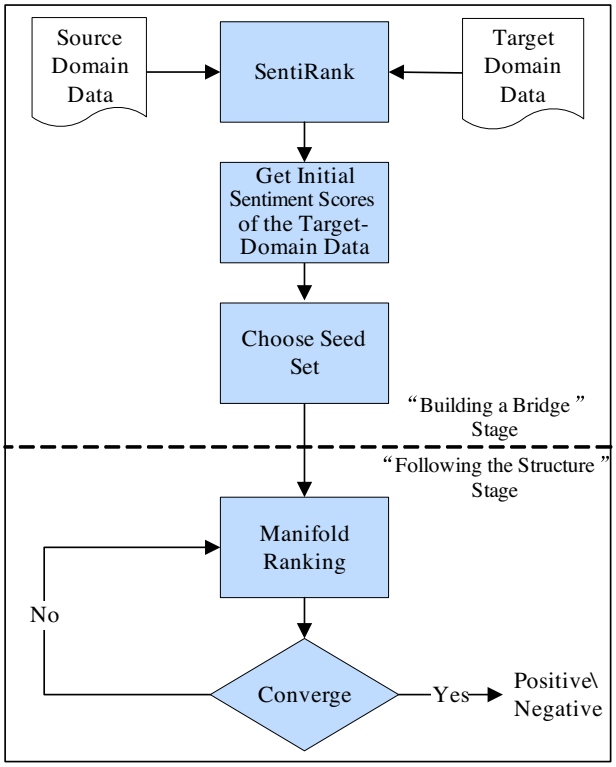

Fig. 1. Framework of the proposed approach.

The algorithms of building a bridge and following the structure are described in details in the next sections, respectively.

3.2. Building a bridge

We begin by making some preprocessing on the datasets. Specifically, we use ICTCLAS (http://ictclas.org/), a Chinese text POS tool, to segment these Chinese reviews. Then, the documents are represented by vector space model. In this model, each document is converted into bag-of-words presentation in the remaining term space, and the term weight is computed with the frequency of the term in the document.

In this step, we take into account these two objections: we aim to (1) get the labels of the target-domain documents while utilizing the information of the source domain, and then (2) identify the high-quality documents which are most confidently labeled.

For the first objection, we use the SentiRank algorithm to build a bridge between the source domain and the target domain.

The SentiRank method (Wu, Tan, & Cheng, 2009) is an algorithm for sentiment transfer and it is used to get the sentiment orientations of the target-domain documents utilizing the similarity between the documents from both the source domain and the target domain. The prior assumption of SentiRank is: if a document is strongly linked with positive (negative) documents, it is probably positive (negative). An intuitive description of SentiRank is as follows: A weighted graph is built from the data, and a sentiment score is assigned for every labeled and unlabeled document to

denote its extent to "negative" or "positive", then the score is iteratively calculated making use of the accurate labels of source-domain data as well as the "pseudo" labels of target-domain data via the weighted graph. The final score for sentiment classification is achieved when the algorithm is converged, so the target-domain data can be labeled based on these scores.

The SentiRank process in our context can be described as follows:

1. In this algorithm, D^U denotes the test data, and D^L denotes the training data; assign every document a sentiment score ("1" denotes positive, and "−1" denotes negative) to represent its degree of sentiment orientation. S^U denotes the sentiment score set of D^U, and S^L denotes the sentiment score set of D^L.
2. Classify D^U with a traditional supervised classifier which is trained by D^L (e.g. prototype classifier (Han & Karypis, 2000), Support Vector Machine (Vapnik, 1998) et al.). Initialize $S^U \cup S^L$ with 1 when the corresponding document is labeled "positive", and with −1 when the corresponding document is labeled "negative". And normalize the sentiment scores to make the sum of positive scores of $D^U(D^L)$ equal to 1, and the sum of negative scores of $D^U(D^L)$ equal to −1.
3. Build a graph whose nodes denote documents in both D^L and D^U and edges denote the content similarities between documents. Create a similarity matrix U from the data points D^L and D^U. Normalize U to $\widehat{U}$ by making the sum of each row equal to 1. Then sort every row of $\widehat{U}$ to $\widetilde{U}$ in descending order in order to find the neighbors of a document, and use a matrix N to denote the neighbors of D^U in source domain. So S^U can be calculated using the sentiment scores of the D^U's neighbors in source domain as follows:

$$s_i^{U(k)} = \sum_{j \in N_{i\bullet}} \left(\widehat{U}_{ij} \times s_j^{L(k-1)} \right), \tag{1}$$

where $i\bullet$ means the ith row of a matrix, $s_i^{U(k)}$ denotes the ith component of S^U at the kth iteration, and $s_j^{L(k-1)}$ denotes the jth component of S^L at the $(k-1)$th iteration.
4. Similarly, a graph is built, in which each node corresponds to a document in D^U and the weight of the edge between any different documents is computed by the cosine measure. Create a similarity matrix V from the data points D^U. V is similarly normalized to $\widehat{V}$ to make the sum of each row equal to 1. Then sort every row of $\widehat{V}$ to $\widetilde{V}$ in descending order to get the neighbors of D^U in the target domain, M. So S^U can be calculated using the sentiment scores of the D^U's neighbors in target domain as follows:

$$s_i^{U(k)} = \sum_{j \in M_{i\bullet}} \left(\widehat{V}_{ij} \times s_j^{U(k-1)} \right). \tag{2}$$

5. In order to make use of the neighbors of D^U in both source domain and target domain, fuse the result of step 3 and step 4, and get the iterative equation as follows to compute S^U at the kth iteration:

$$S^{U(k)} = \alpha \widehat{U} S^{L(k-1)} + \beta \widehat{V} S^{U(k-1)}. \tag{3}$$

In order to converge, normalize S^U at every iteration to make the sum of positive scores of D^U equal to 1, and the sum of negative scores of D^U equal to −1.
6. Iteratively calculate the S^U of D^U and normalize it until it achieves the convergence.
7. According to S^U, assign every document of D^U a label. If the sentiment score is between −1 and 0, assign the document the label "negative"; if the sentiment score is between 0 and 1, assign the document the label "positive".

In this algorithm, α and β show the relative importance of source domain and target domain to the final sentiment scores, and $\alpha + \beta = 1$. According to (Wu et al., 2009), in our experiments, we set α to 0.7 and β to 0.3, which indicates the contribution from source domain is a little more important than that from target domain. Note that the algorithm achieves the convergence when the changing between the sentiment scores computed at two successive iterations for all documents in the target domain falls below a given threshold, and we set the threshold 0.00001 in this work.

Next, we find the high-quality documents from the target domain. To do this, we make use of the sentiment score which denotes its corresponding document's extent to "negative" or "positive". Firstly, we sort the target-domain documents in descending order according to their sentiment scores. So the more forward the document is sorted, the more likely it is positive; the more backward the document is sorted, the more likely it is negative. Then, we choose the first K documents and last K documents as the high-quality documents. In the rest of the paper, we will refer to these high-quality documents as seeds.

In order to prove this algorithm can produce high-quality seeds, we show in Table 1 the labeling accuracies of the K seeds produced by the SentiRank algorithm transferred between our three datasets (see Evaluation Section for details on these datasets). To better evaluate the parameter K, we set K from 50 to 290, an increase of 40 each. Seen from the table, the accuracy decreases gradually with the increase of K. This proves that our approach is effective to sort the target-domain documents according to their opinion extent: the more seeds are chosen, the more noises and uncertainty are involved in the seed set, so the bigger K is, the less accurately the chosen seeds are classified. As we can see, for three tasks (B → H, H → N, N → H), the accuracy is above 89%, and for two tasks (B → N, H → B), the accuracy is above 75%. This high accuracy demonstrates that our approach is well enough to choose high-quality seeds. For the task "N → B", the accuracy is not particularly good. One plausible reason is that the difference between notebook reviews and book reviews is too big for transferring. However, even with imperfectly labeled seeds, we will show in Evaluation Section that our approach can improve the performance of sentiment transfer exploiting these seeds.

3.3. Following the structure

SentiRank algorithm allows us to build a bridge between the source domain and the target domain, but we haven't utilized the distribution of the target domain. In fact, being able to follow the intrinsic structure of the target domain is important for sentiment transfer, as discussed before. Now that we have a small, high-quality seed set which embody the intrinsic structure of the target domain, we can make better use of the seeds by utilizing the manifold-ranking method and having it improve the performance of sentiment transfer.

The manifold-ranking method (Zhou, Weston, Gretton, & SchOlkopf, 2003) is a universal ranking algorithm and it is initially

Table 1
Seed accuracies on six tasks.

Domain	K						
	50	90	130	170	210	250	290
B → H	0.95	0.9222	0.923	0.9294	0.9333	0.934	0.924
B → N	0.82	0.8778	0.8923	0.8912	0.8905	0.882	0.886
H → B	0.80	0.8055	0.8115	0.8117	0.8024	0.754	0.7431
H → N	0.93	0.9277	0.923	0.9235	0.9214	0.91	0.9086
N → B	0.74	0.75	0.7461	0.7264	0.7142	0.712	0.681
N → H	0.9167	0.9111	0.9	0.8976	0.8990	0.898	0.8972

used to rank data points along their underlying manifold structure. The prior assumption of manifold-ranking is: (1) nearby points are likely to have the same ranking scores; (2) points on the same structure (typically referred to as a cluster or a manifold) are likely to have the same ranking scores. An intuitive description of manifold-ranking is as follows: a weighted network is formed on the data, and a positive rank score is assigned to each known relevant point and zero to the remaining points which are to be ranked. All points then spread their ranking score to their nearby neighbors via the weighted network. The spread process is repeated until a global stable state is achieved, and all points obtain their final ranking scores.

Given that we now have a high-quality seed set, we build the weighted network whose points denote documents in D^U. Also, we integrate the sentiment scores of the seeds into the manifold-ranking process. So the sentiment manifold-ranking process can be formalized as follows:

Given a point set $\chi = \{x_1,\ldots,x_K,x_{K+1},\ldots,x_{2K},x_{2K+1},\ldots,x_n\} \subset R^m$, the first K points $x_i (1 \leqslant i \leqslant K)$ are the seeds which are labeled "positive", the second K points $x_j (K+1 \leqslant j \leqslant 2K)$ are the seeds which are labeled "negative", and the remaining points $x_u (2K+1 \leqslant u \leqslant n)$ are unlabeled. Let $F: \chi \to R^2$ denote a ranking function which assigns to each point $x_i (1 \leqslant i \leqslant n)$ a ranking value vector F_i. We can view F as a matrix $F = [F_1^T,\ldots,F_n^T]^T$. We also define a $n \times 2$ matrix $Y = [Y_1, Y_2]$, where $Y_1 = [Y_{11},\ldots,Y_{K1},Y_{K+1,1},\ldots,Y_{n1}]^T$ and $Y_2 = [Y_{12},\ldots,Y_{K2},Y_{K+1,2},\ldots,Y_{n2}]^T$ with $Y_{i1} = 1$ if x_i is labeled as "positive" and $Y_{i2} = 1$ is x_i is labeled as "negative". The manifold ranking algorithm used for sentiment transfer goes as follows:

1. Compute the pair-wise similarity values between points using the cosine measure. The weight associated with term t is calculated with the $tf_t^* idf_t$ formula, where tf_t is the frequency of term t in the document and idf_t is the inverse document frequency of term t, i.e. $1 + \log(N/n_t)$, where N is the total number of documents and n_t is the number of the documents containing term t. Given two points x_i and x_j, the cosine similarity is denoted as $sim(x_i,x_j)$, computed as the normalized inner product of the corresponding term vectors.
2. Connect any two points with an edge if their similarity is not 0. We form the affinity matrix W defined by $W_{ij} = sim(x_i,x_j)$ if $i \neq j$, and we let $W_{ii} = 0$ to avoid loops in the graph built in the next step.
3. Construct the matrix $S = D^{-1/2}WD^{-1/2}$ in which D is a diagonal matrix with its (i,i)-element equal to the sum of the ith row of W.
4. Iterate $F(t+1) = \alpha SF(t) + (1-\alpha)Y$ until convergence, where α is a parameter in $(0,1)$.
5. Let F^* denote the limit of the sequence $\{F(t)\}$. Then every document x_j $(K+1 \leqslant j \leqslant n)$ gets its ranking score vector F_j^*.

In the manifold-ranking algorithm, the weight matrix W is normalized symmetrically in the third step, which is necessary to prove the algorithm's convergence. During the fourth step, every point receives the information from its neighbors (first term), and also retains its initial information (second term). The parameter of manifold-ranking weight α specifies the relative contributions to the ranking scores from its neighbors and its initial ranking scores. According to (Zhou et al., 2003), in our experiment, we set α to 0.6. It is worth mentioning that self-reinforcement is avoided since the diagonal elements of the affinity matrix are set to zero in the second step. Moreover, the information is spread symmetrically since S is a symmetric matrix.

Zhou et al. (2003) proves that the sequence $\{F(t)\}$ converges to

$$F^* = \beta(I - \alpha S)^{-1}Y. \quad (4)$$

Table 2
Accuracy comparison of different methods.

Domain	Proto	TSVM	SentiRank	EM based on Proto	EM based on SentiRank	Manifold based on Proto	Our Approach
B → H	0.735	0.749	0.772	0.765	0.774	0.761	**0.79**
B → N	0.651	0.769	0.714	0.667	0.766	0.745	**0.776**
H → B	0.645	0.614	0.671	**0.723**	0.671	0.677	0.683
H → N	0.729	0.726	0.749	0.657	0.771	0.784	**0.784**
N → B	0.612	0.622	0.638	**0.763**	0.651	0.665	0.65
N → H	0.724	0.772	0.764	0.765	0.777	0.779	**0.791**
Aver	0.683	0.709	0.718	0.723	0.735	0.735	**0.746**

In the above formula, $\beta = 1 - \alpha$. Note that although F^* can be expressed in a closed form, for large scale problems, the iteration algorithm is preferable due to computational efficiency. Usually the convergence of the iteration algorithm is achieved when the difference between the scores computed at two successive iterations for any point falls below a given threshold (0.00001 in this study).

Finally, we label the documents in target domain according to their ranking score vector. For each document $x_i (1 \leqslant i \leqslant n)$, if $Y_{i1} > Y_{i2}$, assign the document the label "positive"; if $Y_{i1} < Y_{i2}$, assign the document the label "negative".

4. Evaluation

Some researchers conducted sentiment-transfer research on English corpus, which are obtained from one web site, and are all product reviews. In order to highlight the domain-specific nature of sentiment expression, we use reviews not only from different web sites, but also from domains with less similarity. Aimed at Chinese applications, we conduct the experiments based on the specialty of Chinese language, and verify the performance on Chinese web reviews. However, the main proposed approach in this paper is language independent in essence.

4.1. Experimental setup

For evaluation, we use three Chinese domain-specific data sets from on-line reviews, which are: Book Reviews[1] (B, from http://www.dangdang.com/), Hotel Reviews[2] (H, from http://www.ctrip.com/) and Notebook Reviews[3] (N, from http://www.360buy.com/). Each dataset has 4000 labeled reviews (2000 positives and 2000 negatives). We choose one of the three data sets as source-domain data, and another data set as target-domain data.

4.2. Baseline systems

In this paper we compare our approach with the following baseline methods:

Proto: This method applies a traditional supervised classifier, prototype classifier (Han & Karypis, 2000), for the sentiment transfer. And it only uses source domain documents as training data. Results of this baseline are shown in column 1 of Table 2. As we can see, accuracy ranges from 61.25% to 73.5%.

TSVM: This method applies transductive SVM (Joachims, 1999) for the sentiment transfer which is a widely used method for improving the classification accuracy. In our experiment, we use Joachims's SVM-light package (http://svmlight.joachims.org/) for TSVM. We use a linear kernel and set all parameters as default. This

[1] www.searchforum.org.cn/tansongbo/corpus/Dangdang_Book_4000.rar.
[2] www.searchforum.org.cn/tansongbo/corpus/Ctrip_htl_4000.rar.
[3] www.searchforum.org.cn/tansongbo/corpus/Jingdong_NB_4000.rar.

method uses both source domain data and target domain data, obtaining the results in column 2 of Table 2. As we can see, accuracy ranges from 61.42% to 77.17%, which are better than Proto.

SentiRank: We implement the SentiRank algorithm where we initialize the sentiment scores by prototype classifier. Table 2 shows the accuracy ranges from 63.7% to 77.2% which are much better than Proto and TSVM.

EM: We implement two versions of the EM algorithm (Dempster, Laird, & Rubin, 1977): one is based on prototype classifier; the other is based on SentiRank algorithm. Taking an example of the EM algorithm based on SentiRank algorithm, specifically, we iteratively train the SentiRank classifier on the data labeled so far, use it to get the sentiment scores of unlabeled documents in the target domain, and augment the labeled data with K_E most confidently labeled documents. Since its performance is highly sensitive to K_E, we test values for K_E from 10 to 300, an increase of 20 each, and reported in column 5 of Table 2 the best results. As we can see, since EM is based on SentiRank, its accuracy ranges from 65.1% to 77.7% which are better than other baselines except Manifold. The EM algorithm based on prototype classifier is similar to the above apart from changing the training classifier from SentiRank to prototype classifier, and its results are in column 4 of Table 2. As we can see, its accuracy ranges from 65.7% to 76.5% which are better than the first three baselines, while worse then the EM algorithm based on SentiRank algorithm.

Manifold: Our last baseline implements the manifold-ranking procedure (Zhou et al., 2003) adaptable for sentiment transfer. Specifically, we begin by training a prototype classifier on the training data. Then we use the similarity scores between the documents and the positive central vector and the similarity scores between the documents and the negative central vector to separately initial the ranking score vectors of the test data. Finally, we choose K_M documents that are most likely to be positive and K_M documents that are most likely to be negative as seeds for manifold-ranking. Since its performance is highly sensitive to K_M, we test values for K_M from 10 to 300, an increase of 20 each, and reported in column 5 of Table 2 the best results. As we can see, accuracy ranges from 66.5% to 78.4% which are better than all other baselines.

4.3. Our approach

In this section, we compare the proposed approach with four baseline methods. There is a parameter, K, in our algorithm. We set K to 290 to show we choose $2K$ seeds for manifold-ranking algorithm. The parameter will be studied in parameter sensitivity section.

Results of our approach are shown in column 7 of Table 2. As we can observe, our approach produces much better performance than all the baselines. The greatest increase of accuracy is achieved by about 12.7% on the problem "H → N" compared to EM based on Proto. The second and third greatest increases of accuracy are achieved by about 12.5% and 6.7% on the problem "B → N" and "N → H" compared to Proto respectively. The greatest average increase of accuracy is achieved by about 6.3% compared to Proto. The great improvement compared with the baselines indicates that our approach performs very effectively and robustly.

Table 2 shows the average accuracies of SentiRank and TSVM are higher than Proto: the average accuracy of SentiRank is about 3.5% higher than Proto, and the average accuracy of TSVM is about 2.6% higher than Proto. As we know, SentiRank and TSVM utilize information of both source domain and target domain while Proto not, so this proves that utilizing the information of two domains is better than utilizing the information of only one domain for improving the accuracy of sentiment transfer.

Seen from Table 2, the average accuracies of the last four columns are higher than the first three columns. The greatest increase of accuracy is achieved by about 6.3% when the 7th column is compared to the 1st column. The least increase of accuracy is achieved by about 0.5% when the 4th column is compared to the 3rd column. As we know, the last four approaches are all two-stage approaches while the first three approaches are not, so this proves that two-stage transfer approach is more effective for sentiment transfer.

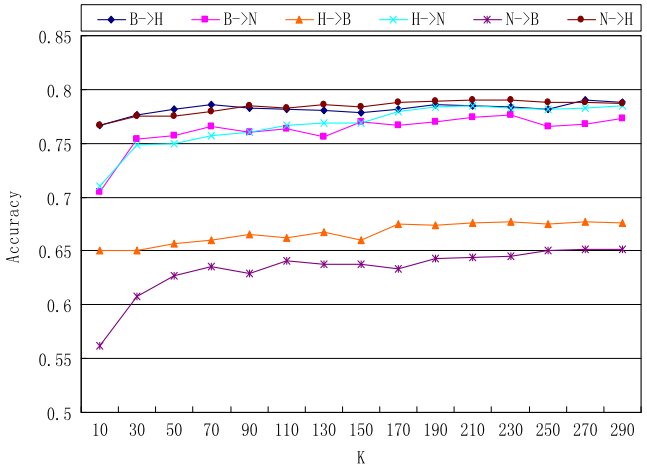

Fig. 2. Accuracy for different K.

Meanwhile, the average accuracy of EM (Manifold) based on SentiRank showed in column 5 (7) is higher than EM (Manifold) based on Proto showed in column 4 (6): the average increase of accuracy of EM based on SentiRank is achieved by about 1.2% compared to EM based on Proto, and the average increase of accuracy of Manifold based on SentiRank is achieved by about 1% compared to Manifold based on Proto, which proves that SentiRank can choose higher quality seeds that embody the intrinsic structure of the domain for next stage.

Table 2 shows the proposed approach outperforms EM: the average accuracy of the proposed approach is about 2.2% higher than EM based on Proto and 1% higher than EM based on SentiRank, and the greatest increase of accuracy is achieved by about 13% on the task "H → N" compared to EM based on Proto. This is caused by two reasons. First, EM is not dedicated for sentiment-transfer learning. Second, our approach can follow the intrinsic structure of the target domain better.

4.4. Parameter sensitivity

In this section, we evaluate the parameter K, and we change K from 10 to 290, an increase of 20 each. We experiment the sensitivity of K on six tasks as mentioned above. And the results are shown in Fig. 2. We can see that the curves of these six examples rise sharply when K increases from 10 to 70, and the curves rise gradually when K increases from 90 to 230, then they become stable after K arrives at 230. This is because when K is too small, the number of seeds is too small to affect the ranking scores of their nearby neighbors, so the performance of our approach is not so good. But when K is bigger than 230, although the number of seeds is big to affect others, the seeds are not accurate enough, and maybe spread some wrong information to their nearby neighbors, so the performance of our approach can not be improved. So we set K to 290 in our overall-performance experiment.

5. Conclusions

We propose a novel two-stage approach for sentiment transfer. Our key idea is to build a bridge between the source domain and

the target domain, and follow the intrinsic structure of the target domain to improve the performance of sentiment transfer. Specifically, we (1) apply the SentiRank algorithm using the accurate labels of source-domain documents as well as the "pseudo" labels of target-domain documents to get the sentiment scores of the target-domain documents, (2) utilize the sentiment scores to identify a small number of most confidently labeled documents as high-quality seeds, (3) employ the manifold-ranking algorithm to spread the seeds' ranking scores to their nearby neighbors to compute the ranking score for every unlabeled document, (4) label the target-domain data based on these scores. Experimental results on three domain-specific sentiment data sets demonstrate that our approach can dramatically improve the accuracy, and can be employed as a high-performance sentiment transfer system.

In future work, we plan to extent our approach to many more domains. Also, since none of the steps in our approach is designed specifically for sentiment classification, we plan to extend our approach to other text classification tasks.

Acknowledgments

This work was mainly supported by two funds, i.e., 60933005 and 60803085.

References

Ando, R., & Zhang, T. (2005). A high-performance semi-supervised learning method for text chunking. In *Proceedings of ACL*.

Andreevskaia, A., & Bergler, S. (2008). When specialists and generalists work together: overcoming domain dependence in sentiment tagging. In *Proceedings of ACL*.

Aue, A., & Gamon, M. (2005). Customizing sentiment classifiers to new domains: A case study. In *Proceedings of RANLP*.

Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification. In *Proceedings of ACL*.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of COLT* (pp. 92–100).

Chawla, N., & Karakoulas, G. (2005). Learning from labeled and unlabeled data: an empirical study across techniques and domains. *Journal of Artificial Intelligence Research, 23*, 331–366.

Chelba, C., & Acero, A. (2004). Adaptation of maximum entropy capitalizer: Little data can help a lot. In *Proceedings of EMNLP*.

Cui, H., Mittal, V., & Datar, M. (2006). Comparative experiments on sentiment classification for online product reviews. In *Proceedings of AAAI*.

Dasgupta, S., & Ng, V. (2009). Mine the easy, classify the hard: A semi-supervised approach to automatic sentiment classification. In *Proceedings of ACL*.

DaumeIII, H., & Marcu, D. (2006). Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research, 26*, 101–126.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, 39*(B), 1–38.

Do, C., & Ng, A. (2005). Transfer learning for text classification. In *Proceedings of NIPS*.

Gamon, M., & Aue, A. (2005). Automatic identification of sentiment vocabulary: Exploiting low association with known sentiment terms. In *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in NLP*.

Han, E., & Karypis, G. (2000). Centroid-based document classification: Analysis & experimental results. In *Proceedings of PKDD*.

Ikeda, D., Takamura, H., & Okumura, M. (2008). Semi-supervised learning for blog classification. In *Proceedings of AAAI*.

Jiang, J., & Zhai, C.X. (2007). A two-stage approach to domain adaptation for Statistical Classifiers. In *Proceedings of CIKM*.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Proceedings of ICML* (pp. 200–209).

Lanquillon, C. (2000). Learning from labeled and unlabeled documents: A comparative study on semi-supervised text classification. In *Proceedings of PKDD* (pp. 490–497).

Luo, P., Zhuang, F., Xiong, H., Xiong, Y., & He, Q. (2008). Transfer learning from multiple source domains via consensus regularization. In *Proceedings of CIKM*.

McDonald, R., Hannan, K., Neylon, T., Wells, M., and Reynar, J. (2007). Structured models for fine-to-coarse sentiment analysis. In Proceedings of ACL.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (1998). Learning to classify text from labeled and unlabeled documents. In *Proceedings of AAAI* pp. 792–799.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques In *Proceedings of EMNLP*.

Raina, R., Battle, A., Lee, H., Packer, B., & Ng, A. (2007). Self-taught learning: Transfer learning from unlabeled data In *Proceedings of ICML*.

Tan, S., Cheng, X., Ghanem, M., Wang, B., & Xu, H. (2005). A novel refinement approach for text categorization. In *Proceedings of CIKM*.

Tan, S. (2005). Neighbor-weighted K-nearest neighbor for unbalanced text corpus. *Expert systems with applications* (Vol. 28, pp. 667–671). Elsevier (4).

Tan, S. (2006). An effective refinement strategy for KNN text classifier. *Expert systems with applications* (Vol. 30, pp. 290–298). Elsevier. 2.

Tan, S., Wang, Y., Wu, G., & Cheng, X. (2008). Using unlabeled data to handle domain-transfer problem of semantic detection. In *Proceedings of SAC*.

Tan, S., Cheng, X., Wang, Y., & Xu, H. (2009). Adapting Naïve Bayes to domain adaptation for sentiment analysis. In *Proceedings of ECIR*.

Vapnik, V. (1998). *Statistical learning theory*. Chichester: Wiley.

Wu, Q., Tan, S., & Cheng, X. (2009). Graph ranking for sentiment transfer. In *Proceedings of ACL*.

Xing, D., Dai, W., Xue, G., & Yu, Y. (2007). Bridged refinement for transfer learning. In *Proceedings of PKDD*.

Zhou, D., Weston, J., Gretton, A., Bousquet O., SchOlkopf, B. (2003). Ranking on data manifolds. In *Proceedings of NIPS*.